



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

**ΤΜΗΜΑ
ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

JAVA SWING

- **ΠΕΡΙΓΡΑΜΜΑΤΑ (BORDERS)**
- **ΕΤΙΚΕΤΕΣ**
- **ΚΟΥΜΠΙΑ**

ΚΕΡΑΜΟΠΟΥΛΟΣ ΕΥΚΛΕΙΔΗΣ

ΠΑΡΑΔΕΙΓΜΑ



ΠΕΡΙΓΡΑΜΜΑΤΑ (BORDERS)

- ⦿ Τα περιγράμματα μπορούν να εμφανιστούν σε κάθε συστατικό της Java Swing.
- ⦿ Με τα περιγράμματα μπορούμε να ορίσουμε εμφανή περιθώρια για κάθε συστατικό.
- ⦿ Η εμφάνιση των περιγραμμάτων είναι ιδιαίτερα χρήσιμη στη χρήση δύο συστατικών (είναι τα μόνα συστατικά των οποίων η εμφάνιση των ορίων δεν ορίζεται αυτόματα):
 - Labels
 - Panels

ΠΕΡΙΓΡΑΜΜΑΤΑ (BORDERS)

- Όλα τα Java Swing Borders είναι στιγμιότυπα της διεπαφής **`javax.swing.border.Border`**
- Μία μεγάλη ποικιλία από μεθόδους υλοποίησης εμφάνισης περιγραμμάτων ενός συστατικού μας παρέχει η Java Swing στην τάξη **`javax.swing.BorderFactory`**
- Παραδείγματα:
 - Δήλωση μεθόδου εμφάνισης απλής γραμμής με επιλογή στο χρώμα
 - `static Border createLineBorder(Color lineColor);`
 - Δήλωση μεθόδου εμφάνισης απλής γραμμής με επιλογή στο χρώμα και στο πάχος
 - `static Border createLineBorder(Color lineColor, int width);`
 - Δήλωση μεθόδου εμφάνισης ανασηκωμένης ή βυθισμένης γραμμής
 - `static Border createEtchedBorder(int type);`
 - Το type παίρνει τιμές: `EtchedBorder.RAISED` ή `EtchedBorder.LOWERED`
 - Δήλωση μεθόδου εμφάνισης κενού γύρω από το συστατικό με επιλογή στο πάχος του κενού σε κάθε πλευρά
 - `static Border createEmptyBorder(int topWidth, int leftWidth, int bottomWidth, int rightWidth);`

Example 1

```
1. import java.awt.*;
2. import javax.swing.*;
3.
4. class BorderDemo {
5.
6.     BorderDemo() {
7.
8.
9.         JFrame jfrm = new JFrame("Use Line and Etched Borders");
10.
11.
12.         jfrm.setLayout(new FlowLayout());
13.
14.
15.         jfrm.setSize(280, 90);
16.
17.
18.         jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19.
20.
21.         JLabel jlab = new JLabel(" This uses a line border. ");
22.         jlab.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

**Ορισμός
ενός line
περιγράμ
ματος**

```
23. JLabel jlab2 = new JLabel(" This uses an etched border. ");
24. jlab2.setBorder(BorderFactory.createEtchedBorder(EtchedBorder.LOWERED));
25.
26. jfrm.add(jlab);
27. jfrm.add(jlab2);
28.
29. jfrm.setVisible(true);
30. }
31.
32. public static void main(String args[]) {
33.     SwingUtilities.invokeLater(new Runnable() {
34.         public void run() {
35.             new BorderDemo();
36.         }
37.     });
38. }
39. }
```



**Ορισμός ενός
etched
περιγράμματος**

ΕΤΙΚΕΤΕΣ

ΕΤΙΚΕΤΕΣ

ΔΟΜΗΤΕΣ	ΠΕΡΙΓΡΑΦΗ
JLabel()	Δημιουργεί μία άδεια ετικέτα
JLabel(String str)	Δημιουργεί μία ετικέτα που περιέχει κείμενο
JLabel(Icon icon)	Δημιουργεί μία ετικέτα που περιέχει εικόνα
JLabel(String str, int horzAlign)	Δημιουργεί μία ετικέτα που περιέχει κείμενο και ορίζει την οριζόντια στοίχιση
JLabel(Icon icon, int horzAlign)	Δημιουργεί μία ετικέτα που περιέχει εικόνα και ορίζει την οριζόντια στοίχιση
JLabel(String str, Icon icon, int horzAlign)	Δημιουργεί μία ετικέτα που περιέχει κείμενο και εικόνα και ορίζει την οριζόντια στοίχιση

ΣΤΟΙΧΙΣΗ ΕΤΙΚΕΤΩΝ

- ◉ Για να στοιχίσουμε **οριζόντια** μία ετικέτα χρησιμοποιούμε τον τέταρτο δομητή, **JLabel(String str, int horzAlign)**
 - Πιθανές τιμές για την παράμετρο horzAlign είναι:

▪ SwingConstants.LEFT	Αριστερά
▪ SwingConstants.RIGHT	Δεξιά
▪ SwingConstants.CENTER	Κέντρο
▪ SwingConstants.LEADING	Είναι αριστερά ή δεξιά ανάλογα την κατεύθυνση του κειμένου (από που ξεκινά)
▪ SwingConstants.TRAILING	Είναι αριστερά ή δεξιά ανάλογα την κατεύθυνση του κειμένου (που τελειώνει)
- ◉ Για να στοιχίσουμε **κάθιστα** μία ετικέτα χρησιμοποιούμε την εντολή **void setVerticalAlignment(int vertAlign);**
 - Πιθανές τιμές για την παράμετρο vertAlign είναι:

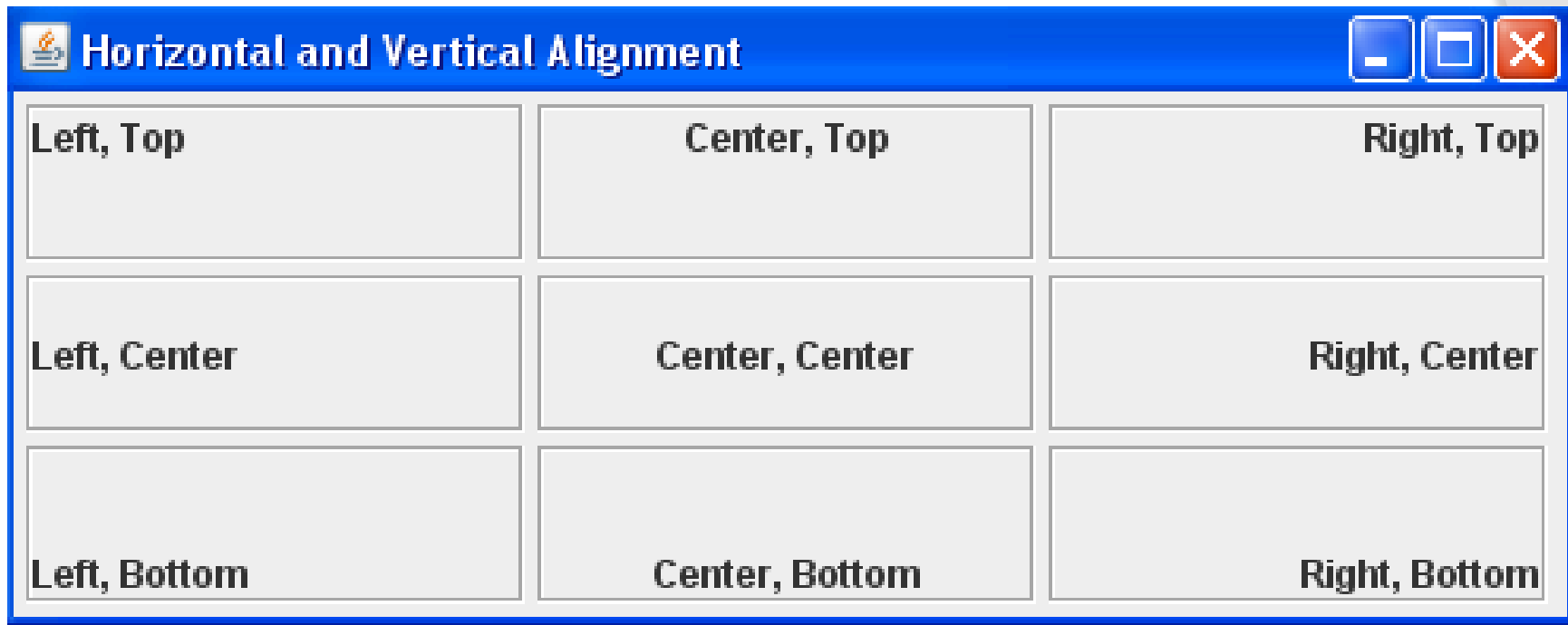
▪ SwingConstants.TOP	Πάνω
▪ SwingConstants.BOTTOM	Κάτω
▪ SwingConstants.CENTER	Κέντρο
- ◉ Η στοίχιση μίας ετικέτας εξαρτάται και από την επιλογή του layout και αν το κείμενο καλύπτει όλο το χώρο της ετικέτας
 - Δεν έχει νόημα στο FlowLayout

GRIDLAYOUT

- ⦿ Αυτή η μορφή layout δημιουργεί έναν πίνακα από κελιά με ίδιο μέγεθος.
- ⦿ `GridLayout(int γραμμές, int στήλες, int οριζόντιο_κενό, int κάθετο_κενό);`
- ⦿ Ο πίνακας γεμίζει από πάνω αριστερά.

ΠΑΡΑΔΕΙΓΜΑ

Επίδειξη στοίχισης κειμένου σε ετικέτες και ορίου σε GridLayout



Example 2

```
1. package ADX_project1;
2. import javax.swing.*;
3. import java.awt.*;
4. import javax.swing.border.*;

5. class AlignLabelDemo {
6.     AlignLabelDemo() {
7.         JLabel[] jlabs = new JLabel[9];
8.         JFrame jfrm = new JFrame("Horizontal and Vertical Alignment");

9.         jfrm.setLayout(new GridLayout(3, 3, 4, 4));

10.        jfrm.setSize(500, 200);
11.        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

To Layout
επιλέχθηκε σε
GridLayout.
Θα αποτελείται
από 3 στήλες και
3 γραμμές με
κενό ανάμεσα
στις στήλες και
τις γραμμές 4
pixels.

```
12. // Left-align text at top.
13. jlabs[0] = new JLabel("Left, Top", SwingConstants.LEFT);
14. jlabs[0].setVerticalAlignment(SwingConstants.TOP);
15. // Center text at top.
16. jlabs[1] = new JLabel("Center, Top", SwingConstants.CENTER);
17. jlabs[1].setVerticalAlignment(SwingConstants.TOP);
18. // Right-align text at Top.
19. jlabs[2] = new JLabel("Right, Top", SwingConstants.RIGHT);
20. jlabs[2].setVerticalAlignment(SwingConstants.TOP);
21. // Left-align text at center.
22. jlabs[3] = new JLabel("Left, Center", SwingConstants.LEFT);
23. // Center the text.
24. jlabs[4] = new JLabel("Center, Center", SwingConstants.CENTER);
25. // Right-align text at center.
26. jlabs[5] = new JLabel("Right, Center", SwingConstants.RIGHT);
27. // Left-align text at bottom..
28. jlabs[6] = new JLabel("Left, Bottom", SwingConstants.LEFT);
29. jlabs[6].setVerticalAlignment(SwingConstants.BOTTOM);
30. // Center text at bottom.
31. jlabs[7] = new JLabel("Center, Bottom", SwingConstants.CENTER);
32. jlabs[7].setVerticalAlignment(SwingConstants.BOTTOM);
33. // Right-align text at bottom.
34. jlabs[8] = new JLabel("Right, Bottom", SwingConstants.RIGHT);
35. jlabs[8].setVerticalAlignment(SwingConstants.BOTTOM);
```

Στις γραμμές 18-42
ορίζονται τα περιεχόμενα
των 9 κελιών.

```
36. Border border = BorderFactory.createEtchedBorder();
37.
38. for(int i=0; i<9; i++)
39.     jlabs[i].setBorder(border);
40.
41. for(int i=0; i<9; i++)
42.     jfrm.add(jlabs[i]);
43.
44. JPanel cp = ((JPanel) jfrm.getContentPane());
45. cp.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));
46. jfrm.setVisible(true);
47.
48. public static void main(String args[]) {
49.     SwingUtilities.invokeLater(new Runnable() {
50.         public void run() {
51.             new AlignLabelDemo();
52.         }
53.     });
54. }
55. }
```

Δημιουργούμε
ένα όριο.

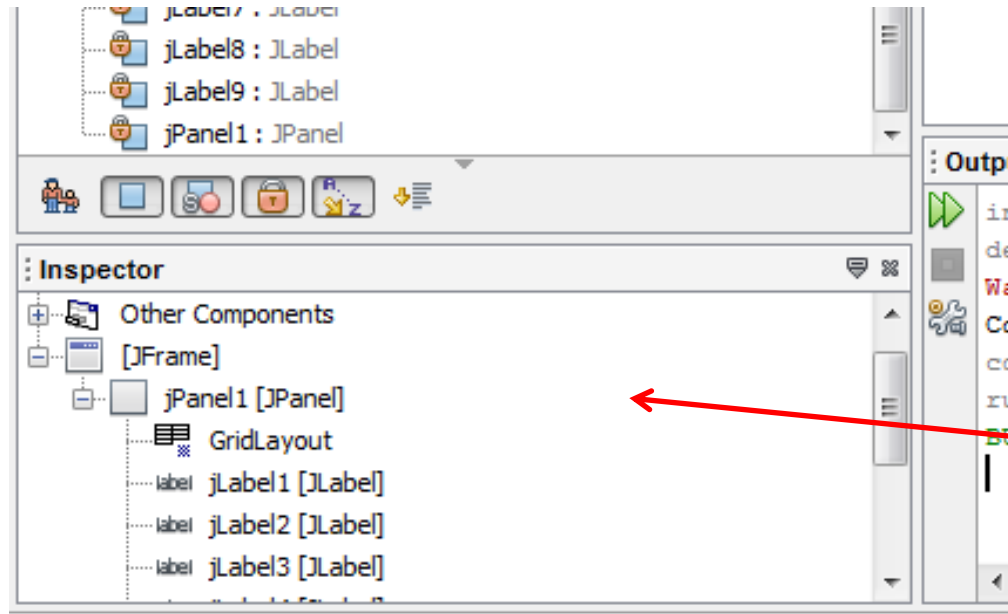
Βάζουμε το όριο
στις 9 ετικέτες

Γεμίζουμε το grid
από πάνω
αριστερά.
Σε περίπτωση
που δεν έχουν
οριστεί 9 ετικέτες
τότε θα μείνουν
άδεια τα κελιά
από κάτω δεξιά.

Προσθέτουμε το
πάνελ σε 2^ο
επίπεδο πάνω
από το frame και
βάζουμε όριο το
«κενό»

ΠΑΡΑΤΗΡΗΣΗ ΓΙΑ ΤΟ NETBEANS

- Δεν υπάρχει ιδιότητα όριο για ένα frame. Έτσι, θα πρέπει πρώτα να βάλουμε πάνω στο frame ένα panel και σε αυτό να ορίσουμε το όριο (border) που θέλουμε και το grid layout.



ΧΡΗΣΗ ΕΙΚΟΝΩΝ ΣΤΙΣ ΕΤΙΚΕΤΕΣ

- Για να προσθέσουμε εικόνα σε μία ετικέτα μπορούμε να χρησιμοποιήσουμε έναν από τους παρακάτω δύο δομητές:
 - JLabel(Icon icon)
 - JLabel(Icon icon, int horzAlign)
- Η εικόνα τοποθετείται στο κέντρο της ετικέτας εκτός και αν την τοποθετήσουμε εμείς σε άλλο σημείο χρησιμοποιώντας την οριζόντια στοίχιση.
- Για να χρησιμοποιήσουμε μία εικόνα σε μία ετικέτα θα πρέπει πρώτα να έχουμε ορίσει την εικόνα ως αντικείμενο της τάξης **javax.swing.ImageIcon** επεκτείνοντας τη διεπαφή **javax.swing.Icon**
 - π.χ Αντιστοιχίζουμε στο αντικείμενο myicon την εικόνα myIcon.gif.
(Η Java δέχεται αρχεία τύπου gif και jpeg για εικόνες)

```
ImageIcon myicon = new ImageIcon("c:/temp/myIcon.gif")
```


ΧΡΗΣΗ ΕΙΚΟΝΩΝ ΚΑΙ ΚΕΙΜΕΝΟΥ ΣΤΙΣ ΕΤΙΚΕΤΕΣ

- ⦿ Για να προσθέσουμε εικόνα και κείμενο σε μία ετικέτα μπορούμε να χρησιμοποιήσουμε τον παρακάτω δομητή:
 - `JLabel(String str, Icon icon, int horzAlign)`
- ⦿ Η εικόνα τοποθετείται αυτόματα στην αριστερή άκρη της ετικέτας και ακολουθεί το κείμενο. Για να επιβάλουμε εμείς διαφορετική τοποθέτηση των δύο αντικειμένων στην ετικέτα θα πρέπει να χρησιμοποιήσουμε τις μεθόδους:
 - `void setVerticalTextPosition(int loc)`
 - `void setHorizontalTextPosition(int loc)`
- ⦿ π.χ
 - `Jlabel1.setVerticalTextPosition(SwingConstants.Top);`

Example 3

```
1. package ADX_project1;
2. import javax.swing.*;
3. import java.awt.*;

4. class IconLabelDemo {

5.     IconLabelDemo() {
6.         JFrame jfrm = new JFrame("Use Images in Labels");
7.         jfrm.setLayout(new GridLayout(4, 1));
8.         jfrm.setSize(250, 300);
9.         jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

4 γραμμές
1 στήλη

10. ImageIcon myIcon = new ImageIcon("myIcon.gif");

Δημιουργία
Αντικειμένου
εικόνας

11.
12. JLabel jLabelIcon = new JLabel(myIcon);

Δημιουργία ετικέτας εικόνας

13.
14. JLabel jLabelIconTxt = new JLabel("Default Icon and Text Position",
15. myIcon, SwingConstants.CENTER);

16. JLabel jLabelIconTxt2 = new JLabel("Text Left of Icon", myIcon,
17. SwingConstants.CENTER);

18. jLabelIconTxt2.setHorizontalTextPosition(SwingConstants.LEFT);

Δημιουργία
ΕΤΙΚΕΤΩΝ
ΕΙΚΟΝΑΣ ΚΑΙ
ΚΕΙΜΕΝΟΥ. Η
στοίχιση είναι
μέσα στο frame

19. JLabel jLabelIconTxt3 = new JLabel("Text Over Icon", myIcon,
SwingConstants.CENTER);

21. jLabelIconTxt3.setVerticalTextPosition(SwingConstants.TOP);

22. jLabelIconTxt3.setHorizontalTextPosition(SwingConstants.CENTER);

Το κείμενο να
μπει αριστερά

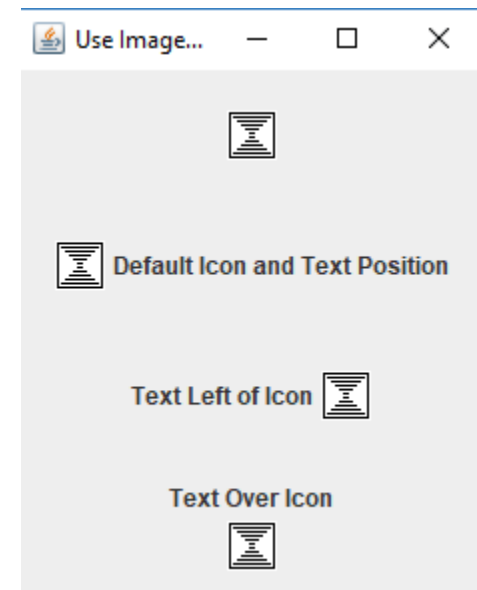
Το κείμενο να
μπει πάνω

```

23.    jfrm.add(jlblIcon);
24.    jfrm.add(jlblIconTxt);
25.    jfrm.add(jlblIconTxt2);
26.    jfrm.add(jlblIconTxt3);
27.    jfrm.setVisible(true);
28. }

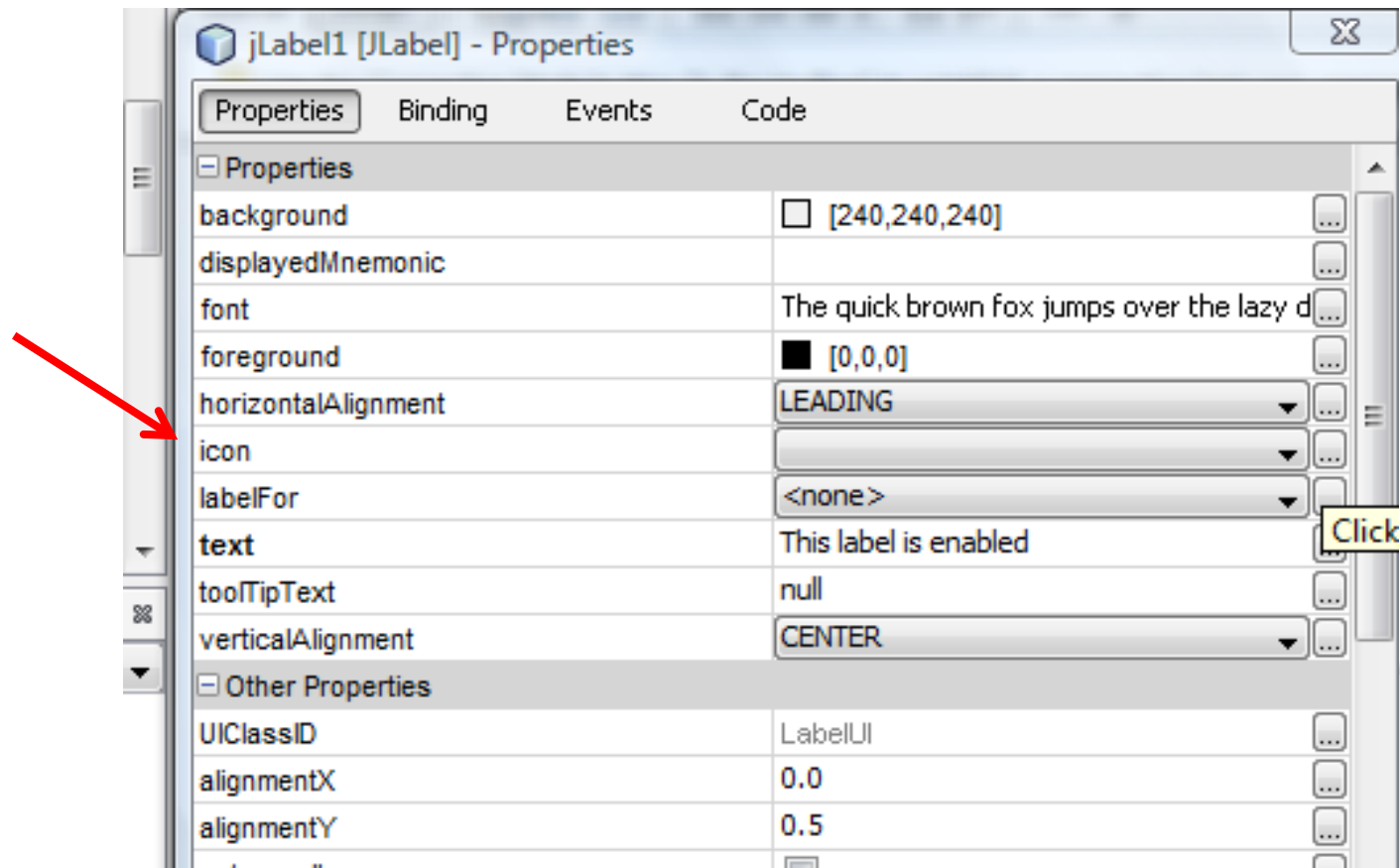
29. public static void main(String args[]) {
30.     SwingUtilities.invokeLater(new Runnable() {
31.         public void run() {
32.             new IconLabelDemo();
33.         }
34.     });
35. }
36. }

```



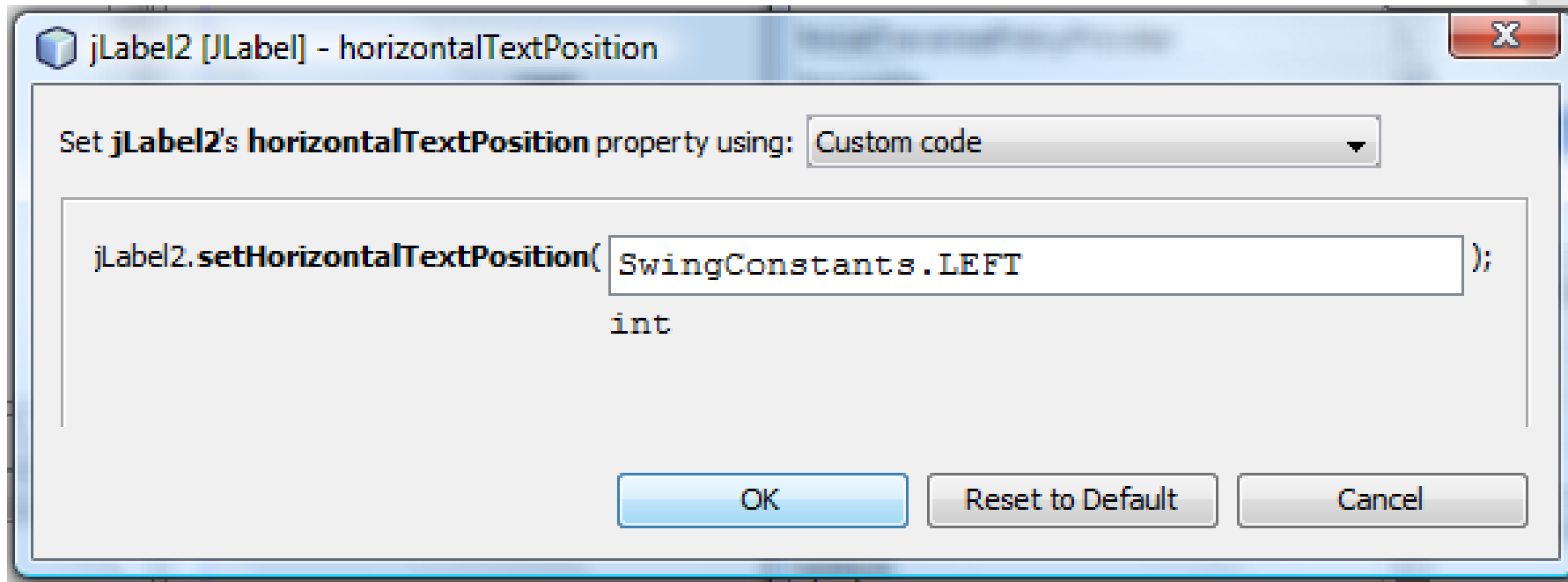
ΠΩΣ ΒΑΖΩ ΕΙΚΟΝΑ ΣΤΟ NETBEANS

- Επιλέγω την ιδιότητα icon του συστατικού π.χ JLabel, ...



ΣΤΟΙΧΙΣΗ ΚΕΙΜΕΝΟΥ ΕΤΙΚΕΤΑΣ ΜΕ ΕΙΚΟΝΑ ΣΤΟ NETBEANS

Επιλέγω την ιδιότητα `horizontalTextPosition` της ετικέτας



ΟΙ ΕΤΙΚΕΤΕΣ ΜΠΟΡΟΥΝ ΝΑ ΕΜΦΑΝΙΣΤΟΥΝ «ΑΠΕΝΕΡΓΟΠΟΙΗΜΕΝΕΣ»

- Οι ετικέτες χρησιμοποιούνται συνήθως για να περιγράψουν κάποιο άλλο συστατικό, όπως κάποιο πεδίο κειμένου.
 - Σε περίπτωση που αυτό το συστατικό είναι απενεργοποιημένο αισθητικά και λειτουργικά είναι καλύτερα να φαίνεται και η ετικέτα απενεργοποιημένη (γκρι).
 - Αυτό γίνεται με την μέθοδο
void setEnabled(boolean κατάσταση)
η **κατάσταση** μπορεί να είναι true (ενεργοποιημένη) ή false (απενεργοποιημένη)
 - Σε περίπτωση που στην ετικέτα υπάρχει εικόνα μπορεί να αλλάζει η εικόνα για φαίνεται η ετικέτα απενεργοποιημένη
 - Αυτό γίνεται με την μέθοδο
void setDisabledIcon(Icon εικόνα)

Example 4

```
1. package ADX_project1;
2. import javax.swing.*;
3. import java.awt.*;

4. class DisabledLabelDemo {
5.     DisabledLabelDemo() {
6.         JFrame jfrm = new JFrame("Use Images in Labels");

7.         jfrm.setLayout(new GridLayout(3, 1));

8.         jfrm.setSize(240, 250);

9.         jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```


- ```
10. ImageIcon myIcon = new ImageIcon("myIcon.gif");
11. ImageIcon myDisIcon = new ImageIcon("myDisIcon.gif");

12. JLabel jLabelIconTxt = new JLabel("This label is enabled.", myIcon, SwingConstants.CENTER);

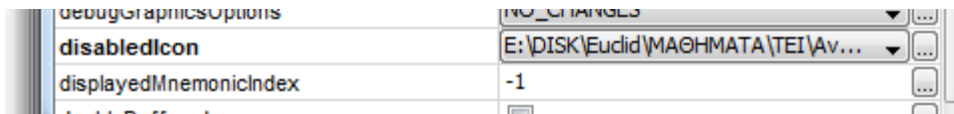
13. JLabel jLabelIconTxt2 = new JLabel("This label is disabled.", myIcon, SwingConstants.CENTER);
14. jLabelIconTxt2.setEnabled(false);
```



Απενεργοποίηση  
της ετικέτας με  
αφαίρεση του v

Απενεργοποίηση  
της ετικέτας

- ```
15. JLabel jLabelIconTxt3 = new JLabel("Use the disabled icon.", myIcon, SwingConstants.CENTER);
16. jLabelIconTxt3.setDisabledIcon(myDisIcon);
17. jLabelIconTxt3.setEnabled(false);
```

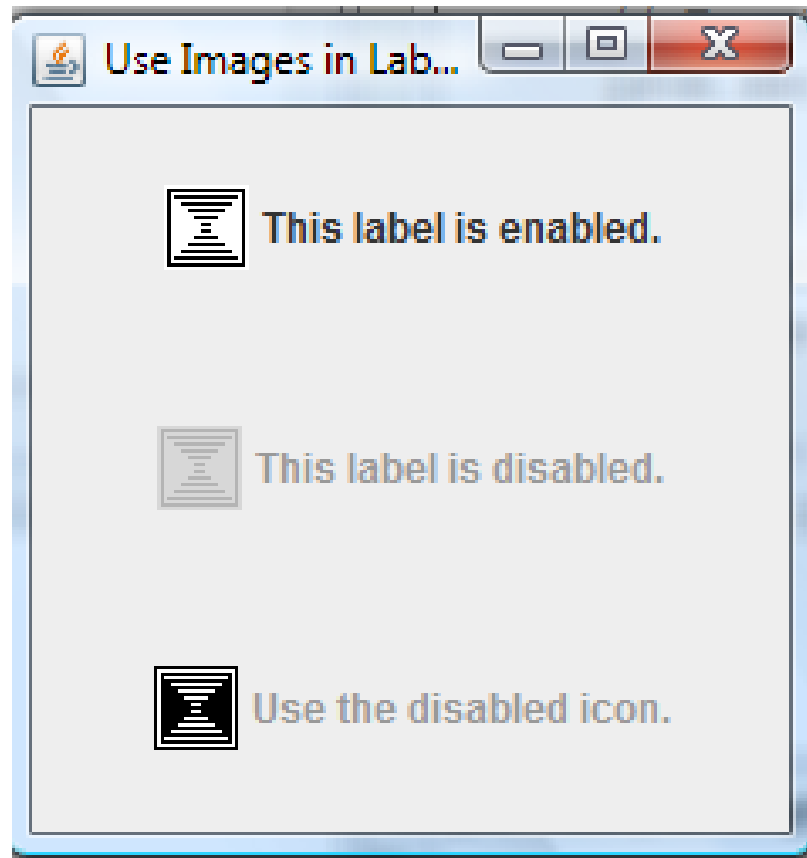


Ορισμός εικόνας σε
κατάσταση
απενεργοποίηση της
ετικέτας

```
1.    jfrm.add(jlblconTxt);
2.    jfrm.add(jlblconTxt2);
3.    jfrm.add(jlblconTxt3);

4.    jfrm.setVisible(true);
5.    }

6.    public static void main(String args[]) {
7.        SwingUtilities.invokeLater(new Runnable() {
8.            public void run() {
9.                new DisabledLabelDemo();
10.            }
11.        });
12.    }
13. }
```



ΟΡΙΣΜΟΣ ΣΥΝΤΟΜΕΥΣΗΣ (SHORTCUT KEY) ΣΕ ΕΤΙΚΕΤΕΣ

- Υπάρχει η δυνατότητα να οριστεί συντόμευση για γρήγορη πρόσβαση σε ένα συστατικό μέσω της ετικέτας που το περιγράφει.
- Αυτό γίνεται σε δύο βήματα:

- Ορίζεται το πλήκτρο συντόμευσης με τη μέθοδο

void setDisplayedMnemonic(int πλήκτρο);

π.χ **label1.setDisplayedMnemonic('f');**

Για την ετικέτα label1 ορίστηκε πλήκτρο συντόμευσης το 'f'.

Σε περίπτωση που εμφανίζεται στην ετικέτα πολλές φορές το πλήκτρο συντόμευσης τότε υπογραμμίζεται η πρώτη εμφάνιση. Αν όμως θέλουμε κάποια άλλη εμφάνιση του ίδιου πλήκτρου τότε χρησιμοποιούμε τη παρακάτω μέθοδο:

void setDisplayedMnemonicIndex(int θέση) throws IllegalArgumentException;

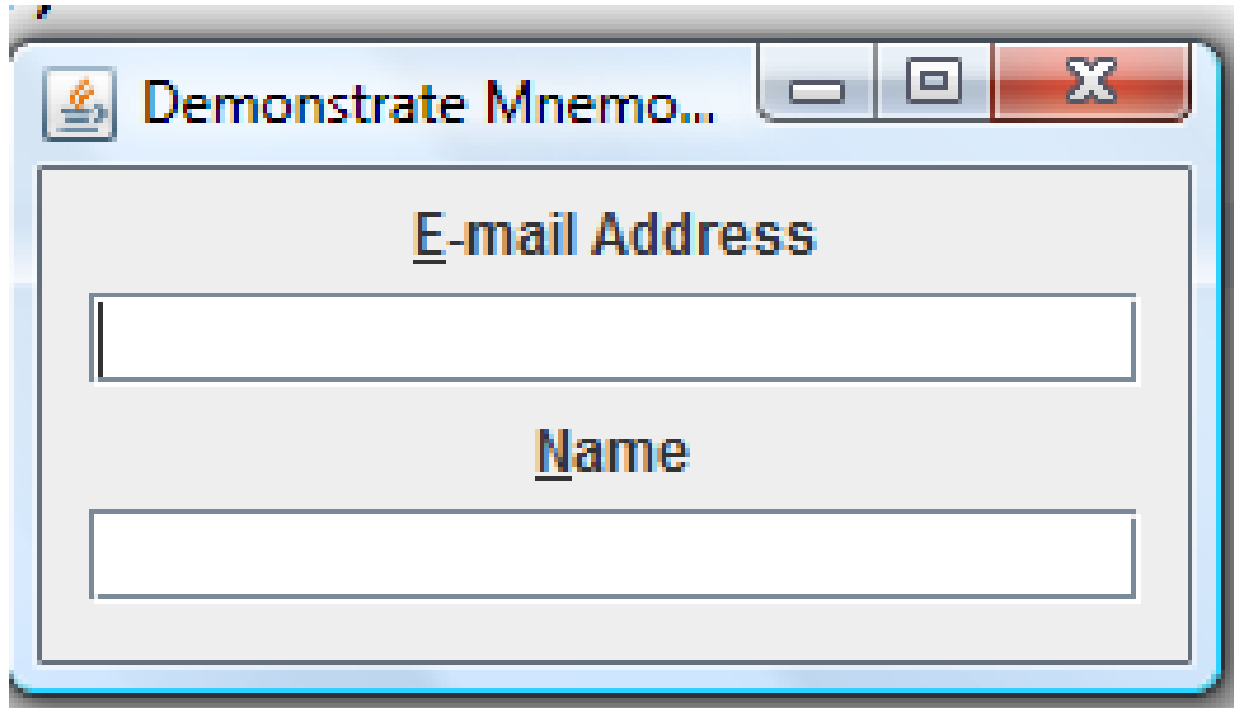
π.χ **label1.setDisplayedMnemonicIndex(9);**

- Ορίζεται το συστατικό που συνδέεται στην ετικέτα με τη μέθοδο

void setLabelFor(Component συστατικό);

π.χ. **label1.setLabelFor(textfield1)**

ΠΑΡΑΔΕΙΓΜΑ



A screenshot of a Java Swing window titled "Demonstrate Mnemo...". The window has a standard Mac OS X-style title bar with a red close button, a yellow maximize button, and a green minimize button. Inside the window, there is a light gray panel containing two text input fields. The first field is labeled "E-mail Address" and the second field is labeled "Name". Both labels are centered above their respective input fields.

Example 5

```
1. package ADX_project1;
2. import java.awt.*;
3. import javax.swing.*;

4. class MnemDemo {
5.     MnemDemo() {
6.         JFrame jfrm = new JFrame("Demonstrate Mnemomics");
7.         jfrm.setLayout(new FlowLayout());
8.         jfrm.setSize(260, 140);
9.         jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10.        JLabel jlab1 = new JLabel("E-mail Address");
11.        JLabel jlab2 = new JLabel("Name");
```

12. jlab1.setDisplayedMnemonic('e');
13. jlab2.setDisplayedMnemonic('n');

Ορίζονται για κάθε
ετικέτα οι συντομεύσεις

14. JTextField jtf1 = new JTextField(20);
15. JTextField jtf2 = new JTextField(20);

Δημιουργούνται δύο
Text Field

16. jlab1.setLabelFor(jtf1);
17. jlab2.setLabelFor(jtf2);

Ορίζεται ποιο συστατικό
αντιστοιχεί σε ποια
ετικέτα

```
18.    jfrm.add(jlab1);
19.    jfrm.add(jtf1);
20.    jfrm.add(jlab2);
21.    jfrm.add(jtf2);
22.    jfrm.setVisible(true);
23.    }

24.    public static void main(String args[]) {
25.        SwingUtilities.invokeLater(new Runnable() {
26.            public void run() {
27.                new MnemDemo();
28.            }
29.        });
30.    }
31. }
```


ΧΡΗΣΙΜΕΣ ΜΕΘΟΔΟΙ ΓΙΑ ΕΤΙΚΕΤΕΣ

- Αλλαγή του κειμένου μιας ετικέτας γίνεται με τη μέθοδο:
void setText(String str);
- Παίρνουμε το κείμενο μίας ετικέτας με τη μέθοδο:
String getText();
- Μπορεί να χρησιμοποιηθεί HTML για τη περιγραφή του κειμένου.
π.χ
`JLabel label1 = new JLabel("<html>Top
Bottom");`

ΚΟΥΜΠΙΑ

ΚΟΥΜΠΙΑ (BUTTONS)

- Στη Java Swing υπάρχουν τέσσερα είδη κουμπιών:
 1. JButton, το κλασσικό κουμπί (push-button)
 2. JToggleButton, ένα κουμπί δύο καταστάσεων (on/off)
 3. JCheckBox, κουμπί επιλογής
 4. JRadioButton, κουμπί μοναδικής επιλογής από ένα σετ κουμπιών.
- Όλα τα κουμπιά είναι υποτάξεις της AbstractButton η οποία επεκτείνει την JComponent.
- Στην AbstractButton είναι υλοποιημένες πολλές μέθοδοι οι οποίες μπορούν να εφαρμοσθούν σε όλα τα είδη κουμπιών.

ΜΕΘΟΔΟΙ ΤΗΣ ABSTRACTBUTTON

ΜΕΘΟΔΟΣ	ΠΕΡΙΓΡΑΦΗ
<code>void addActionListener(ActionListener al)</code>	Καταχωρεί έναν ακροατή γεγονότων ενέργειας σχετικό με κουμπί
<code>void addChangeListener(ChangeListener cl)</code>	Καταχωρεί έναν ακροατή γεγονότων μεταβολής σχετικό με κουμπί
<code>void addItemListener(ItemListener il)</code>	Καταχωρεί έναν ακροατή γεγονότων κατάστασης σχετικό με κουμπί
<code>void doClick()</code>	Αλλάζει ένα κουμπί σε πατημένο (pressed)
<code>String getText()</code>	Παίρνει το κείμενο από ένα κουμπί
<code>ButtonModule getModel()</code>	Επιστρέφει το μοντέλο ενός κουμπιού
<code>boolean isSelected()</code>	Επιστρέφει true αν το κουμπί είναι επιλεγμένο ή false σε αντίθετη περίπτωση (εφαρμόζεται μόνο σε JToggleButton)
<code>void setEnabled(boolean state)</code>	Ενεργοποιεί και Απενεργοποιεί ένα κουμπί.
<code>void setDisabledIcon(Icon icon)</code>	Όταν το κουμπί είναι απενεργοποιημένο θέτει την εικόνα του
<code>void setHorizontalAlignment(int HorzAlign)</code>	Στοιχίζει οριζόντια τα περιεχόμενα ενός κουμπιού

ΜΕΘΟΔΟΙ ΤΗΣ ABSTRACTBUTTON (2)

ΜΕΘΟΔΟΣ	ΠΕΡΙΓΡΑΦΗ
<code>void setHorizontalTextPosition(int loc)</code>	Στοιχίζει οριζόντια το κείμενο ενός κουμπιού το οποίο περιέχει και εικόνα και κείμενο
<code>void setMnemonic(int ch)</code>	Δημιουργεί συντόμευση χαρακτήρα για ένα κουμπί
<code>void setPressedIcon(Icon icon)</code>	Όταν το κουμπί είναι πατημένο θέτει την εικόνα του
<code>void setRolloverIcon(Icon icon)</code>	Όταν το ποντίκι περνά πάνω από ένα κουμπί θέτει την εικόνα του
<code>void setRolloverSelectedIcon(Icon icon)</code>	Όταν το κουμπί είναι επιλεγμένο και περνά από πάνω του το ποντίκι θέτει την εικόνα του
<code>void setSelected(boolean state)</code>	Αλλάζει την κατάσταση ενός κουμπιού. Με true το κουμπί γίνεται επιλεγμένο ενώ με false επιστρέφει στην αρχική του κατάσταση
<code>void setSelectedIcon(Icon icon)</code>	Όταν το κουμπί είναι επιλεγμένο θέτει την εικόνα του
<code>void setText(String str)</code>	Ορίζει το κείμενο σε ένα κουμπί
<code>void setVerticalAlignment(int vertAlign)</code>	Στοιχίζει κάθετα τα περιεχόμενα ενός κουμπιού
<code>void setVerticalTextPosition(int loc)</code>	Στοιχίζει κάθετα το κείμενο ενός κουμπιού το οποίο περιέχει και εικόνα και κείμενο

ΚΑΤΗΓΟΡΙΕΣ ΓΕΓΟΝΟΤΩΝ ΚΟΥΜΠΙΩΝ

- ◎ Υπάρχουν τριών ειδών κατηγορίες γεγονότων:
 - Action γεγονότα
 - Item γεγονότα
 - Change γεγονότα

ΓΕΓΟΝΟΤΑ ACTION

- ◉ Όταν μία ενέργεια εκτελείται τότε δημιουργείται ένα αντίστοιχο γεγονός.
 - π.χ. το πάτημα ενός κουμπιού.
- ◉ Η **ActionEvent** τάξη περιγράφει αυτού του είδους τα γεγονότα, τα οποία χειρίζεται η μοναδική μέθοδος της διεπαφής **ActionListener**:
 - `void actionPerformed(ActionEvent ae);`

ΓΕΓΟΝΟΤΑ ACTION

◎ Πως λειτουργεί;

- Το αντικείμενο `ActionEvent` που δημιουργείται, στέλνεται στη μέθοδο `actionPerformed()`, με την οποία έχουμε πρόσβαση σε διάφορες πληροφορίες.
- Βρίσκουμε το συστατικό που δημιούργησε το γεγονός. Αυτό γίνεται με δύο μεθόδους:
 - `String getActionCommand()`,
επιστρέφει το `string` που περιέχει το κουμπί που δημιούργησε το γεγονός
 - `Object getSource()`,
επιστρέφει μία αναφορά στο κουμπί που δημιούργησε το γεγονός

ΓΕΓΟΝΟΤΑ ACTION

⦿ Χρήσιμες μέθοδοι

- `int getModifiers()`,
επιστρέφει αν κάποιο ειδικό πλήκτρο έχει πατηθεί, όπως ALT, CTRL, SHIFT, ...
- `long getWhen()`,
επιστρέφει τη στιγμή που έγινε το γεγονός σε μορφή timestamp.

Σημείωση: Timestamp αποτελείται από
ΕΤΟΣ:ΜΗΝΑ:ΜΕΡΑ:ΩΡΑ:ΛΕΠΤΟ:ΔΕΥΤΕΡΟΛΕΠΤΟ:ΔΕΚΑΤΟ

ΓΕΓΟΝΟΤΑ ITEM

- Τα γεγονότα κατάστασης δημιουργούνται όταν συμβαίνει ένα γεγονός σε ένα **JCheckBox** ή σε ένα **JRadioButton**.
- Η **ItemEvent** τάξη περιγράφει αυτού του είδους τα γεγονότα, τα οποία χειρίζεται η μοναδική μέθοδος της διεπαφής **ItemListener**:
 - void itemStateChanged(ItemEvent ie);
- Για να πάρουμε μία αναφορά στο συστατικό που συνέβη το γεγονός χρησιμοποιούμε τη μέθοδο:
 - Object getItem(),
η οποία επιστρέφει ένα αντικείμενο τύπου JCheckBox ή JRadioButton
- Για να πάρουμε ποια συστατικά μπορούν να δημιουργήσουν ένα ItemEvent γεγονός χρησιμοποιούμε τη μέθοδο:
 - ItemSelectable getItemSelectable(),
- Για να μάθουμε την τιμή που έχει ένα συστατικό ενός JCheckBox ή JRadioButton χρησιμοποιούμε τη μέθοδο:
 - int getStateChange(),
που παίρνει τιμές SELECTED και DESELECTED

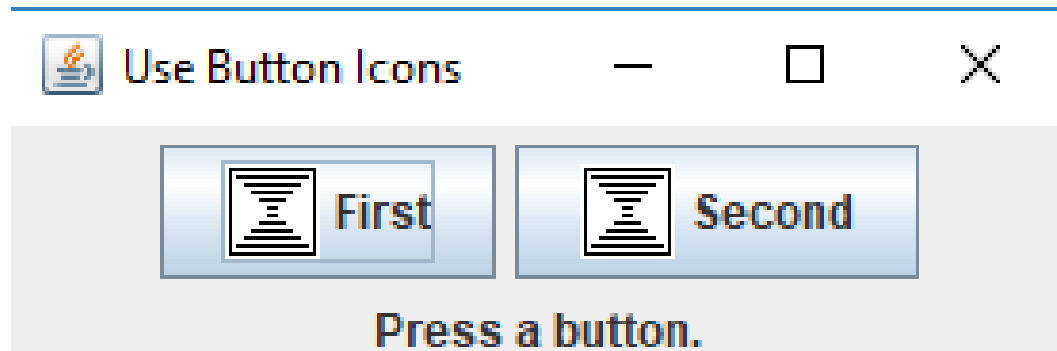
ΓΕΓΟΝΟΤΑ CHANGE

- ◉ Τα γεγονότα μεταβολής κατάστασης δημιουργούνται όταν συμβαίνει ένα γεγονός σε ένα **JToggleButton**.
- ◉ Η **ChangeEvent** τάξη περιγράφει αυτού του είδους τα γεγονότα, τα οποία χειρίζονται η μοναδική μέθοδος της διεπαφής **ChangeListener**:
 - `void stateChanged(ChangeEvent ce);`
- ◉ Για να πάρουμε ένα συστατικό που δημιουργήθηκε από **ChangeEvent** γεγονός χρησιμοποιούμε τη μέθοδο
 - `Object getSource();`

JBUTTON – ΔΟΜΗΤΕΣ

- ◉ Ένα JButton μπορεί να περιέχει **κείμενο**, **εικόνα** ή και τα δύο.
- ◉ Ένα JButton δημιουργείται με έναν από τους παρακάτω δομητές:
 - JButton(),
επιστρέφει ένα κενό κουμπί
 - JButton(String str)
επιστρέφει ένα κουμπί που εμφανίζει το κείμενο str
 - JButton(String str, Icon icon)
επιστρέφει ένα κουμπί που εμφανίζει το κείμενο str και την εικόνα icon
 - JButton(Icon icon)
επιστρέφει ένα κουμπί που εμφανίζει την εικόνα icon
 - JButton(Action act)
επιστρέφει ένα κουμπί που τα περιεχόμενα του εξαρτώνται από το αντικείμενο παράμετρο

JBUTTON



JBUTTON – ΠΑΡΑΔΕΙΓΜΑ

```
1. package ADX_project1;
2. import java.awt.*;
3. import java.awt.event.*;
4. import javax.swing.*;
5. class ButtonIcons implements ActionListener {
6.     JLabel jlab;
7.     JButton jbtnFirst;
8.     JButton jbtnSecond;

9.     ButtonIcons() {
10.         ImageIcon myIcon = new ImageIcon("myIcon.gif");
11.         ImageIcon myDisIcon = new ImageIcon("myDisIcon.gif");
12.         ImageIcon myROIcon = new ImageIcon("myROIcon.gif");
13.         ImageIcon myPIcon = new ImageIcon("myPIcon.gif");

14.         JFrame jfrm = new JFrame("Use Button Icons");
15.         jfrm.setLayout(new FlowLayout());
16.         jfrm.setSize(220, 100);
17.         jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Example 6

Ορίζονται οι εικόνες που
θα χρησιμοποιηθούν

18. jlab = new JLabel("Press a button.");

19. jbtnFirst = new JButton("First", myIcon);

20. jbtnSecond = new JButton("Second", myIcon);

21. jbtnFirst.setRolloverIcon(myROIcon);

22. jbtnSecond.setRolloverIcon(myROIcon);

23. jbtnFirst.setPressedIcon(myPIcon);

24. jbtnSecond.setPressedIcon(myPIcon);

25. jbtnFirst.setDisabledIcon(myDisIcon);

26. jbtnSecond.setDisabledIcon(myDisIcon);

27. jbtnFirst.addActionListener(this);

28. jbtnSecond.addActionListener(this);

29. jfrm.add(jbtnFirst);

30. jfrm.add(jbtnSecond);

31. jfrm.add(jlab);

32. jfrm.setVisible(true);

33. }

Δημιουργία δύο νέων κουμπιών
που περιέχουν κείμενο και εικόνα

Ποια εικόνα θα εμφανίζεται σε
κάθε κουμπί όταν γίνεται rollover

Ποια εικόνα θα εμφανίζεται σε
κάθε κουμπί όταν επιλέγεται

Ποια εικόνα θα εμφανίζεται σε κάθε
κουμπί όταν απενεργοποιείται

Καταχωρούμε τον ακροατή για το
αντίστοιχο γεγονός για κάθε κουμπί

```
34. public void actionPerformed(ActionEvent ae) {
35.     if(ae.getActionCommand().equals("First")) {
36.         jlab.setText("First button was pressed.");
37.         if(jbtnSecond.isEnabled()) {
38.             jlab.setText("Second button is disabled.");
39.             jbtnSecond.setEnabled(false);
40.         }
41.         else {
42.             jlab.setText("Second button is enabled.");
43.             jbtnSecond.setEnabled(true);
44.         }
45.     } else
46.         jlab.setText("Second button was pressed. ");
47. }

48. public static void main(String args[]) {
49.     SwingUtilities.invokeLater(new Runnable() {
50.         public void run() {
51.             new ButtonIcons();
52.         }
53.     });
54. }
55. }
```

Αν το second κουμπί
είναι ενεργοποιημένο
απενεργοποιείται αλλιώς
το αντίστροφο

JBUTTON ΧΡΗΣΙΜΕΣ ΜΕΘΟΔΟΙ

- ◎ void setDefaultButton(JButton button)
ορίζει ένα κουμπί ως default. Δηλαδή όταν πατήσουμε το πλήκτρο ENTER τότε θα «πατηθεί» αυτόματα το default κουμπί.
 - π.χ.
`Jfrm.getRootPane().setDefaultButton(jbtnFirst);`
- ◎ Μπορούμε να χρησιμοποιήσουμε
 - HTML σε ένα κουμπί όπως και στις ετικέτες και επίσης
 - να ορίσουμε συντομεύσεις με τη μέθοδο `setMnemonic()` όπως και στις ετικέτες

ΔΕΥΤΕΡΗ ΛΥΣΗ ΜΕ ΧΡΗΣΗ ANONYMOUS INNER CLASSES

```
1. package ADX_project1;
2. import java.awt.*;
3. import java.awt.event.*;
4. import javax.swing.*;
5. class ButtonIcons implements ActionListener {
6.     JLabel jlab;
7.     JButton jbtnFirst;
8.     JButton jbtnSecond;

9.     ButtonIcons() {
10.         ImageIcon myIcon = new ImageIcon("c:/temp/myIcon.gif");
11.         ImageIcon myDisIcon = new ImageIcon("c:/temp/myDisIcon.gif");
12.         ImageIcon myROIcon = new ImageIcon("c:/temp/myROIcon.gif");
13.         ImageIcon myPIcon = new ImageIcon("c:/temp/myPIcon.gif");

14.         JFrame jfrm = new JFrame("Use Button Icons");
15.         jfrm.setLayout(new FlowLayout());
16.         jfrm.setSize(220, 100);
17.         jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Η τάξη ButtonIcons δεν
επεκτείνει τη διεπαφή
ActionListener

```
18.   jlab = new JLabel("Press a button.");

19.   jbtnFirst = new JButton("First", myIcon);
20.   jbtnSecond = new JButton("Second", myIcon);

21.   jbtnFirst.setRolloverIcon(myROIcon);
22.   jbtnSecond.setRolloverIcon(myROIcon);

23.   jbtnFirst.setPressedIcon(myPIcon);
24.   jbtnSecond.setPressedIcon(myPIcon);

25.   jbtnFirst.setDisabledIcon(myDisIcon);
26.   jbtnSecond.setDisabledIcon(myDisIcon);

27.   jbtnFirst.addActionListener(this);
28.   jbtnSecond.addActionListener(this);

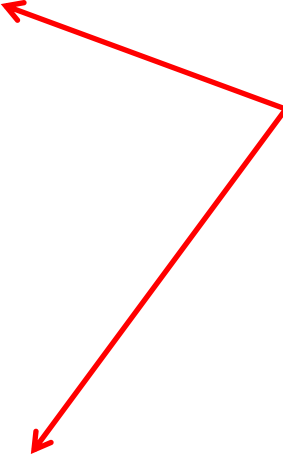
29.   jfrm.add(jbtnFirst);
30.   jfrm.add(jbtnSecond);
31.   jfrm.add(jlab);
32.   jfrm.setVisible(true);
```



Δεν καταχωρούμε τον ακροατή για το αντίστοιχο γεγονός για κάθε κουμπί

```
33. jbtnFirst.addActionListener(new ActionListener() {
34.     public void actionPerformed(ActionEvent ae) {
35.         if(jbtnSecond.isEnabled()) {
36.             jlab.setText("Second button is disabled.");
37.             jbtnSecond.setEnabled(false);
38.         } else {
39.             jlab.setText("Second button is enabled.");
40.             jbtnSecond.setEnabled(true);
41.         }
42.     }
43. });
44.
45. jbtnSecond.addActionListener(new ActionListener() {
46.     public void actionPerformed(ActionEvent ae) {
47.         jlab.setText("Second button was pressed.");
48.     }
49. });
50. public static void main(String args[]) {
51.     SwingUtilities.invokeLater(new Runnable() {
52.         public void run() {
53.             new ButtonIcons();
54.         }
55.     });
56. }
57. }
```

Η υλοποίηση του χειρισμού των γεγονότων γίνεται από μία ανώνυμη εσωτερική τάξη για κάθε περίπτωση. Στο NetBeans χρησιμοποιούνται ανώνυμες εσωτερικές τάξεις στον κώδικα που δημιουργείται αυτόματα.



ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ BUTTONMODEL

Example 7

```
1. package ADX_project1;

2. import java.awt.*;
3. import javax.swing.*;
4. import javax.swing.event.*;

5. class ChangeDemo {
6.     JButton jbtn;
7.     JLabel jlab;
8.     ChangeDemo() {
9.         JFrame jfrm = new JFrame("Button Change Events");
10.        jfrm.setLayout(new FlowLayout());
11.        jfrm.setSize(250, 160);
12.        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13.        jlab = new JLabel();
14.        jbtn = new JButton("Press for Change Event Test");
```

Το μοντέλο ButtonModel
περιέχει έτοιμες καταστάσεις
που μπορούμε να
χρησιμοποιήσουμε

```
15. jbtn.addChangeListener(new ChangeListener() {
16.     public void stateChanged(ChangeEvent ce) {
17.         ButtonModel mod = jbtn.getModel();
18.         String what = "";

19.         if(mod.isEnabled()) what += "Enabled<br>";
20.         if(mod.isRollover()) what += "Rollover<br>";
21.         if(mod.isArmed()) what += "Armed<br>";
22.         if(mod.isPressed()) what += "Pressed<br>";
23.         jlab.setText("<html>Current state:<br>" + what);
24.     }
25. });
26. jfrm.add(jbtn);
27. jfrm.add(jlab);
28. jfrm.setVisible(true);
29. }

33. public static void main(String args[]) {
34.     SwingUtilities.invokeLater(new Runnable() {
35.         public void run() {
36.             new ChangeDemo();
37.         }
38.     });
39. }
40. }
```

Δημιουργείται ανώνυμη εσωτερική
τάξη για διαχείριση των γεγονότων
μεταβολής στο κουμπί jbtn

Επέκταση του μοντέλου
ButtonModel

Χρήση HTML για την εμφάνιση

JTOGGLEBUTTON

- Το JToggleButton είναι ένα JButton το οποίο όταν «πατιέται» μένει μέσα. Έτσι έχει δύο καταστάσεις
 1. pressed
 2. released
- Το JToggleButton είναι υπερτάξη του JCheckBox και του JRadioButton, που σημαίνει ότι στην τάξη JToggleButton περιγράφεται ένα μεγάλο κομμάτι της λειτουργικότητας και των άλλων δύο τύπων κουμπιών.
- Το JToggleButton χρησιμοποιεί ένα διαφορετικό μοντέλο από το JButton, το **JToggleButton.JToggleButtonModel**
- Μία άλλη διαφορά που έχει το JToggleButton από το JButton είναι ότι το JToggleButton χειρίζεται επίσης και γεγονότα ItemEvents

JTOGGLEBUTTON - ΔΟΜΗΤΕΣ

- Όλα τα Toggle κουμπιά που δημιουργούνται είναι αντικείμενα της τάξης JToggleButton.
- Ένα Toggle κουμπί δημιουργείται με έναν από τους παρακάτω δομητές:
 - JToggleButton()
Δημιουργεί ένα κουμπί χωρίς περιεχόμενο (κείμενο ή εικόνα)
 - JToggleButton(String str)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str
 - JToggleButton(Icon icon)
Δημιουργεί ένα κουμπί που εμφανίζει μία εικόνα icon
 - JToggleButton(String str, boolean state)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str δίνοντας και την αρχική κατάσταση του
 - JToggleButton(Icon icon, boolean state)
Δημιουργεί ένα κουμπί που εμφανίζει μία εικόνα icon δίνοντας και την αρχική κατάσταση του
 - JToggleButton(String str, Icon icon)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str και την εικόνα icon
 - JToggleButton(String str, Icon icon, boolean state)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str και την εικόνα icon και την αρχική κατάσταση του
 - JToggleButton(Action act)
Δημιουργεί ένα κουμπί όπου η εμφάνιση του και η κατάσταση του εισάγονται από το αντικείμενο act

ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ TOGGLEBUTTON

```
1. package ADX_project1;
2. import java.awt.*;
3. import java.awt.event.*;
4. import javax.swing.*;

5. class ToggleDemo {
6.     JLabel jlab;
7.     JToggleButton jtbn;
8.     ToggleDemo() {
9.         JFrame jfrm = new JFrame("Demonstrate a Toggle Button");
10.        jfrm.setLayout(new FlowLayout());
11.        jfrm.setSize(290, 80);
12.        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13.        jlab = new JLabel("Button is off.");
14.        jtbn = new JToggleButton("On/Off");
```

Example 8

Δημιουργία Toggle button

```
15.  jtbn.addItemListener(new ItemListener() {
16.      public void itemStateChanged(ItemEvent ie) {
17.          if(jtbn.isSelected())
18.              jlab.setText("Button is on.");
19.          else
20.              jlab.setText("Button is off.");
21.      }
22.  });

23.  jfrm.add(jtbn);
24.  jfrm.add(jlab);
25.  jfrm.setVisible(true);
26.  }

27.  public static void main(String args[]) {
28.      SwingUtilities.invokeLater(new Runnable() {
29.          public void run() {
30.              new ToggleDemo();
31.          }
32.      });
33.  }
34.  }
```

Διαχείριση των γεγονότων
αλλαγής κατάστασης του
κουμπιού jtbn με ανώνυμη
εσωτερική τάξη

ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΔΥΟ TOGGLEBUTTON

Example 9

```
1. package ADX_project1;
2. import java.awt.*;
3. import java.awt.event.*;
4. import javax.swing.*;


5. class TwoTBDemo implements ItemListener {
6.     JLabel jlabAlpha;
7.     JLabel jlabBeta;
8.     JToggleButton jtbnAlpha;
9.     JToggleButton jtbnBeta;
10.    TwoTBDemo() {
11.        JFrame jfrm = new JFrame("Two Toggle Buttons");
12.        jfrm.setLayout(new FlowLayout());
13.        jfrm.setSize(290, 80);
14.        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15.        jlabAlpha = new JLabel("Alpha is off. ");
16.        jlabBeta = new JLabel("Beta is off.");
```

```
17.     jtbnAlpha = new JToggleButton("Alpha");
18.     jtbnBeta = new JToggleButton("Beta");


19.     jtbnAlpha.addItemListener(this);
20.     jtbnBeta.addItemListener(this);

21.     jfrm.add(jtbnAlpha);
22.     jfrm.add(jlabAlpha);
23.     jfrm.add(jtbnBeta);
24.     jfrm.add(jlabBeta);
25.     jfrm.setVisible(true);
26. }

27. public void itemStateChanged(ItemEvent ie) {
28.     JToggleButton tb = (JToggleButton) ie.getItem();
29.     if(tb == jtbnAlpha)
30.         if(tb.isSelected())
31.             jlabAlpha.setText("Alpha is on. ");
32.         else
33.             jlabAlpha.setText("Alpha is off. ");
34.     else if(tb == jtbnBeta)
35.         if(tb.isSelected())
36.             jlabBeta.setText("Beta is on.");
37.         else
38.             jlabBeta.setText("Beta is off.");
39. }
```



Πρόσθεση ακροατών για
διαχείριση γεγονότων
κατάστασης



Η μέθοδος getItem()
επιστρέφει το αντικείμενο που
δημιούργησε το γεγονός

```
40. public static void main(String args[]) {  
41.     SwingUtilities.invokeLater(new Runnable() {  
42.         public void run() {  
43.             new TwoTBDemo();  
44.         }  
45.     });  
46. }  
47. }
```

JCHECKBOX - ΔΟΜΗΤΕΣ

- ◉ Ένα JCheckBox κουμπί δημιουργείται με έναν από τους παρακάτω δομητές:
 - JCheckBox()
Δημιουργεί ένα κουμπί χωρίς περιεχόμενο (κείμενο ή εικόνα)
 - JCheckBox(String str)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str
 - JCheckBox(Icon icon)
Δημιουργεί ένα κουμπί που εμφανίζει μία εικόνα icon
 - JCheckBox(String str, boolean state)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str δίνοντας και την αρχική κατάσταση του
 - JCheckBox(Icon icon, boolean state)
Δημιουργεί ένα κουμπί που εμφανίζει μία εικόνα icon δίνοντας και την αρχική κατάσταση του
 - JCheckBox(String str, Icon icon)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str και την εικόνα icon
 - JCheckBox(String str, Icon icon, boolean state)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str και την εικόνα icon και την αρχική κατάσταση του
 - JCheckBox(Action act)
Δημιουργεί ένα κουμπί όπου η εμφάνιση του και η κατάσταση του εισάγονται από το αντικείμενο act

JCHECKBOX

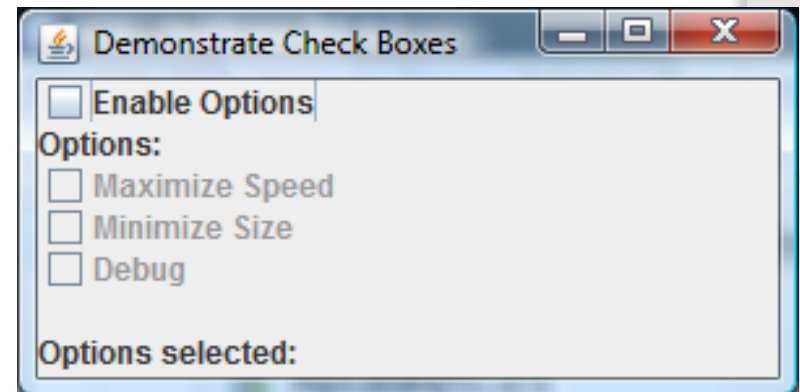
- Το CheckBox κουμπί (όπως και όλα τα κουμπιά καταστάσεων) μπορεί να αλλάξει κατάσταση είτε προγραμματιστικά είτε από την επίδραση του χρήστη.
- Προγραμματιστικά θα πρέπει να χρησιμοποιηθεί η μέθοδος
void setSelected(boolean state)
- Όλα τα CheckBox κουμπιά που δημιουργούνται είναι αντικείμενα της τάξης JCheckBox.

ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CHECKBOX

```
1. package ADX_project1;
2. import java.awt.*;
3. import java.awt.event.*;
4. import javax.swing.*;

5. class CBDemo implements ItemListener {
6.     JLabel jlabOptions;
7.     JLabel jlabWhat;
8.     JLabel jlabChange;
9.     JCheckBox jcbOptions;
10.    JCheckBox jcbSpeed;
11.    JCheckBox jcbSize;
12.    JCheckBox jcbDebug;
```

Example 10




```
13. CBDemo() {
14.     JFrame jfrm = new JFrame("Demonstrate Check Boxes");
15.     jfrm.setLayout(new GridLayout(7, 1));
16.     jfrm.setSize(300, 150);
17.     jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18.     jlabOptions = new JLabel("Options:");
19.     jlabChange = new JLabel("");
20.     jlabWhat = new JLabel("Options selected:");
21.     jcbOptions = new JCheckBox("Enable Options");
22.     jcbSpeed = new JCheckBox("Maximize Speed");
23.     jcbSize = new JCheckBox("Minimize Size");
24.     jcbDebug = new JCheckBox("Debug");
25.     jcbSpeed.setEnabled(false);
26.     jcbSize.setEnabled(false);
27.     jcbDebug.setEnabled(false);
```

Τα τρία checkbox
δημιουργούνται
απενεργοποιημένα

```
28. jcbOptions.addItemListener(new ItemListener() {
29.     public void itemStateChanged(ItemEvent ie) {
30.         if(jcbOptions.isSelected()) {
31.             jcbSpeed.setEnabled(true);
32.             jcbSize.setEnabled(true);
33.             jcbDebug.setEnabled(true);
34.         }
35.         else {
36.             jcbSpeed.setEnabled(false);
37.             jcbSize.setEnabled(false);
38.             jcbDebug.setEnabled(false);
39.         }
40.     }
41. });
```

Ελέγχεται το checkbox
“option”. Αλλάζει την
κατάσταση του και ανάλογα
με την τιμή του ενεργοποιεί ή
απενεργοποιεί τα άλλα τρία
checkboxes

```
42. jcbSpeed.addItemListener(this);
43. jcbSize.addItemListener(this);
44. jcbDebug.addItemListener(this);
```

Ορίζονται οι ακροατές για τα
τρία checkbox

```
45. jfrm.add(jcbOptions);
46. jfrm.add(jlabOptions);
47. jfrm.add(jcbSpeed);
48. jfrm.add(jcbSize);
49. jfrm.add(jcbDebug);
50. jfrm.add(jlabChange);
51. jfrm.add(jlabWhat);
52. jfrm.setVisible(true);
53. }
```

```

54. public void itemStateChanged(ItemEvent ie) {
55.     String opts = "";
56.     JCheckBox cb = (JCheckBox) ie.getItem();
57.     if(ie.getStateChange() == ItemEvent.SELECTED)
58.         jlabChange.setText("Selection change: " + cb.getText() + " selected.");
59.     else
60.         jlabChange.setText("Selection change: " + cb.getText() + " cleared.");
61.     if(jcbSpeed.isSelected()) opts += "Speed ";
62.     if(jcbSize.isSelected()) opts += "Size ";
63.     if(jcbDebug.isSelected()) opts += "Debug ";
64.     jlabWhat.setText("Options selected: " + opts);
65. }

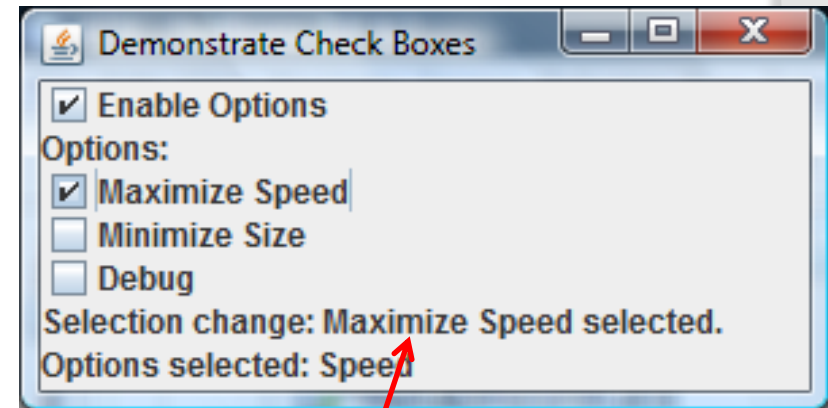
```

Ελέγχεται ποιο
check box
επιλέχθηκε και
εμφανίζεται το
κατάλληλο
μήνυμα

```

66. public static void main(String args[]) {
67.     SwingUtilities.invokeLater(new Runnable() {
68.         public void run() {
69.             new CBDemo();
70.         }
71.     });
72. }
73. }

```



Η cb.getText() εμφανίζει "Maximize Speed"

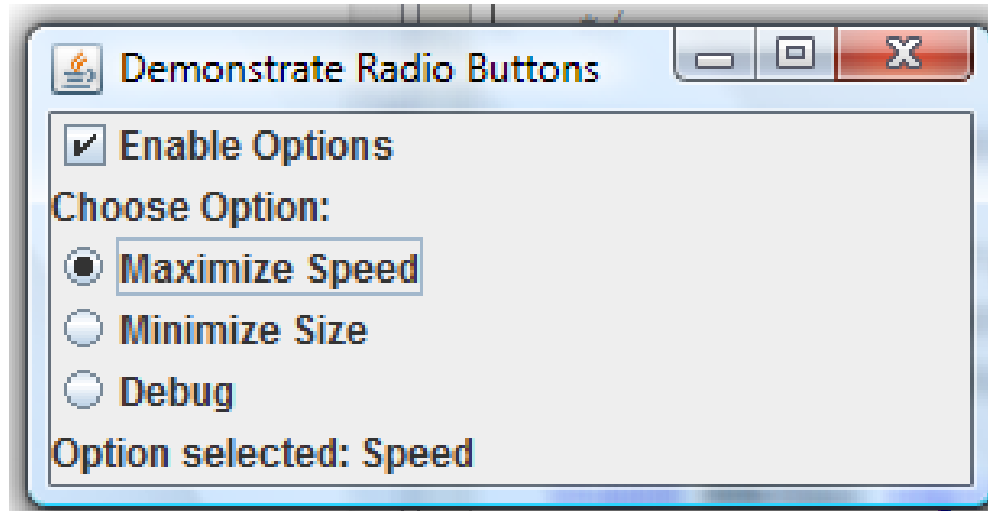
JRADIOBUTTON

- Στη περίπτωση του JRadioButton από ένα σύνολο επιλογών μόνο μία απάντηση μπορεί να επιλεγεί.
- Χρησιμοποιώντας την τάξη ButtonGroup μπορούμε να δημιουργήσουμε σετ από radiobutton.
- Όλα τα RadioButton που δημιουργούνται είναι αντικείμενα της τάξης JRadioButton.
- Κάθε JRadioButton δημιουργεί μία επιλογή από το σετ

RADIOBUTTON- ΔΟΜΗΤΕΣ

- ◉ Ένα JRadioButton κουμπί δημιουργείται με έναν από τους παρακάτω δομητές:
 - JRadioButton()
Δημιουργεί ένα κουμπί χωρίς περιεχόμενο (κείμενο ή εικόνα)
 - JRadioButton(String str)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str
 - JRadioButton(Icon icon)
Δημιουργεί ένα κουμπί που εμφανίζει μία εικόνα icon
 - JRadioButton(String str, boolean state)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str δίνοντας και την αρχική κατάσταση του
 - JRadioButton(Icon icon, boolean state)
Δημιουργεί ένα κουμπί που εμφανίζει μία εικόνα icon δίνοντας και την αρχική κατάσταση του
 - JRadioButton(String str, Icon icon)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str και την εικόνα icon
 - JRadioButton(String str, Icon icon, boolean state)
Δημιουργεί ένα κουμπί που εμφανίζει το κείμενο str και την εικόνα icon και την αρχική κατάσταση του
 - JRadioButton(Action act)
Δημιουργεί ένα κουμπί όπου η εμφάνιση του και η κατάσταση του εισάγονται από το αντικείμενο act

ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ RADIOBUTTON



```
1. package ADX_project1;
2. import java.awt.*;
3. import java.awt.event.*;
4. import javax.swing.*;
5. class RBDemo implements ActionListener {
6.     JLabel jlabOptions;
7.     JLabel jlabWhat;
8.     JCheckBox jcbOptions;
9.     JRadioButton jrbSpeed;
10.    JRadioButton jrbSize;
11.    JRadioButton jrbDebug;
```

Example 11

```
13. RBDemo() {
14.     JFrame jfrm = new JFrame("Demonstrate Radio Buttons");
15.     jfrm.setLayout(new GridLayout(6, 1));
16.     jfrm.setSize(300, 150);
17.     jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18.     jlabOptions = new JLabel("Choose Option:");
19.     jlabWhat = new JLabel("Option selected: Speed");
20.     jcbOptions = new JCheckBox("Enable Options");
21.     jrbSpeed = new JRadioButton("Maximize Speed", true);
22.     jrbSize = new JRadioButton("Minimize Size");
23.     jrbDebug = new JRadioButton("Debug");
```

```
24.     ButtonGroup bg = new ButtonGroup();
25.     bg.add(jrbSpeed);
26.     bg.add(jrbSize);
27.     bg.add(jrbDebug);
```

```
28.     jrbSpeed.setEnabled(false);
29.     jrbSize.setEnabled(false);
30.     jrbDebug.setEnabled(false);
```

Τα τρία radiobutton μπαίνουν
στην ίδια ομάδα

Τα τρία radiobutton
δημιουργούνται
απενεργοποιημένα

```
31. jcbOptions.addItemListener(new ItemListener() {
32.     public void itemStateChanged(ItemEvent ie) {
33.         if(jcbOptions.isSelected()) {
34.             jrbSpeed.setEnabled(true);
35.             jrbSize.setEnabled(true);
36.             jrbDebug.setEnabled(true);
37.         }
38.         else {
39.             jrbSpeed.setEnabled(false);
40.             jrbSize.setEnabled(false);
41.             jrbDebug.setEnabled(false);
42.         }
43.     }
44. });
```

Ελέγχεται το checkbox
"option". Αλλάζει την
κατάσταση του και ανάλογα
με την τιμή του ενεργοποιεί ή
απενεργοποιεί τα άλλα τρία
radiobutton

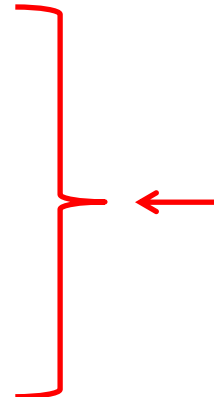

```
45. jrbSpeed.addActionListener(this);
46. jrbSize.addActionListener(this);
47. jrbDebug.addActionListener(this);
```



Ορίζονται οι ακροατές για τα
τρία radiobutton

```
48. jfrm.add(jcbOptions);
49. jfrm.add(jlabOptions);
50. jfrm.add(jrbSpeed);
51. jfrm.add(jrbSize);
52. jfrm.add(jrbDebug);
53. jfrm.add(jlabWhat);
54. jfrm.setVisible(true);
55. }
```

```
56. public void actionPerformed(ActionEvent ie) {
57.     String opts = "";
58.     if(jrbSpeed.isSelected()) opts = "Speed ";
59.     else if(jrbSize.isSelected()) opts = "Size ";
60.     else opts = "Debug";
61.     jlabWhat.setText("Option selected: " + opts);
62. }
```

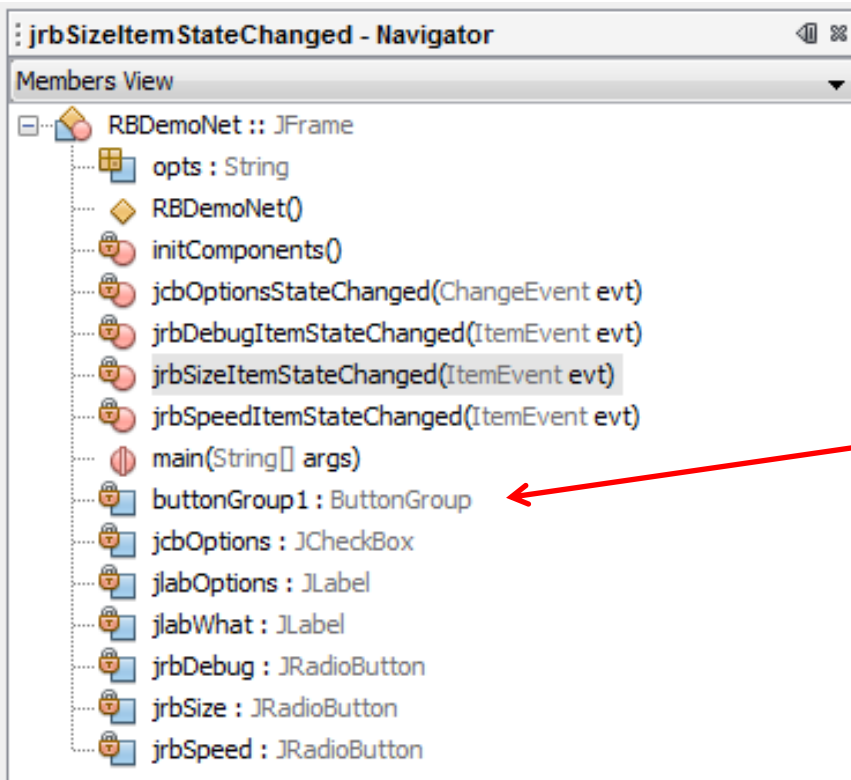


Ελέγχεται ποιο
radiobutton επιλέχθηκε
και εμφανίζεται το
κατάλληλο μήνυμα

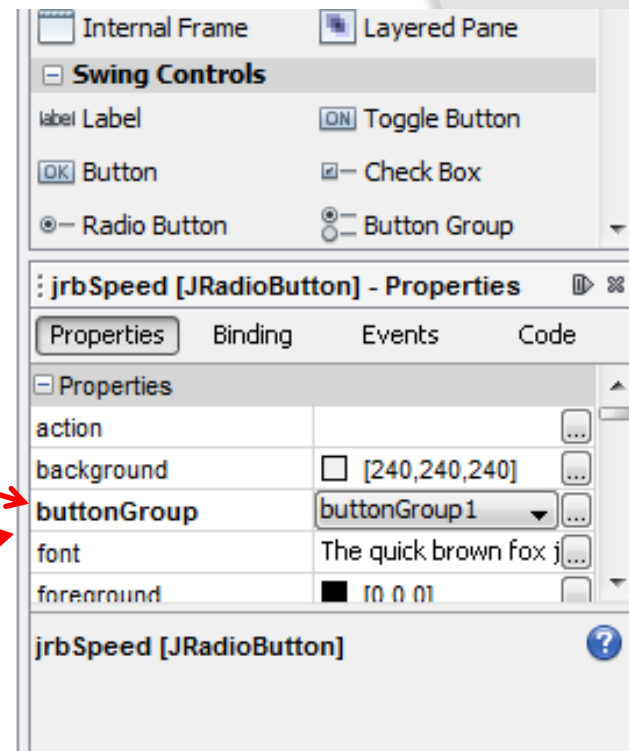
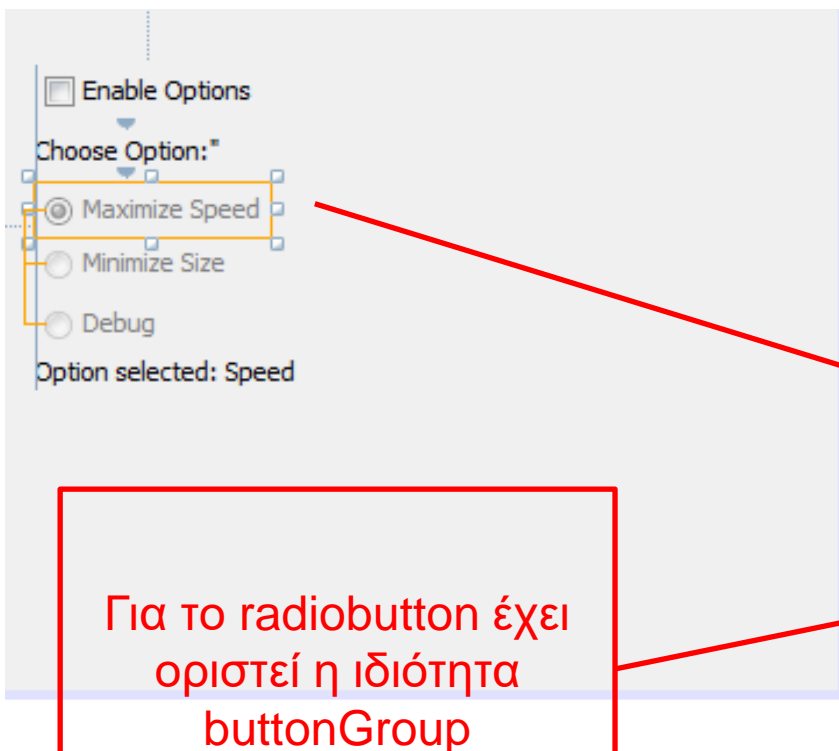
```

63. public static void main(String args[]) {
64.     SwingUtilities.invokeLater(new Runnable() {
65.         public void run() {
66.             new RBDemo();
67.         }
68.     });
69. }
70. }

```



ΠΡΟΣΟΧΗ
Θα πρέπει όλα τα
radiobutton να τα
βάλετε στην ίδια ομάδα
κουμπιών για να
λειτουργήσουν σωστά.



επόμενη ενότητα

JAVA SWING

- SCROLL BARS
- SLIDERS
- PROGRESS BARS